GridFTP Improvements

Igor Mandrichenko, FNAL

GGF8, Seattle, June 24-27 2003

Background and History

- GridFTP Working Group produced GridFTP v1.0 protocol draft document
- The Working Group was re-chartered
  - Instead of discontinuing, it was decided to go on with further development of GridFTP protocol
- Goal: improve GridFTP protocol as bulk file-oriented data transfer protocol for grid applications
  - Reliability, robustness
  - Efficiency, performance
  - Flexibility
  - Backward compatibility with RFC959 and other IETF standards

List of issues and improvements for v2.0

- Between GGF7 and GGF8 mailing list and WG web page were created and used to communicate and finalize the list of points of improvement
  - Mailing list: gridftp-wg@gridforum.org
  - Web site: http://www-isd.fnal.gov/gridftp-wg
    - To be moved under GridForge (http://forge.gridforum.org)
- Goal for today: to finalize the list of issues (not including concrete solutions!)
- Shortly after GGF8: submit the list as GGF document

Issues: Unidirectional transfers in E mode

- Data must flow in the same direction as data socket connection initiation
- Uploads must be passive, downloads must be active
  - Will not work with firewalls, private networks, etc.
- Reason: possible race condition leading to lost data connections

Issues: Order of PASV/SPAS and STOR/RETR commands

- In passive mode, server must provide data socket address *before* it knows what to transfer
- Inherited from RFC959 FTP protocol
- Difficult to implement distributed server

Issues: Possible disconnection of idle control and data sockets

- Some firewalls drop idle TCP connections

- Server with cache/staging functionality may need to have control and even data channels stay open and idle for “long” time

Issues: Unreliable EOF communication in Stream mode

- Per RFC959: closing of data channel signals end of data

- Server can not distinguish between client termination and end of successful upload transfer

Issues: Control over server feedback

- GridFTP server periodically sends 1xx responses during transfer (performance markers)
- Can be used to protect control channel against being timed-out by the firewall
- Client needs to have control over frequency and contents of the feedback

Issues: Data protection

- TCP offers transmission error detection with checksums
- May not be sufficient for high volumes of data
- The idea is to introduce additional means of protection against transmission and storage errors

Issues: adoption of IETF draft for structured LIST

- There is IETF proposal for “ls”-like commands with structured output designed to be parsed by the client
  - ftp://ftp.isi.edu/internet-drafts/draft-ietf-ftpext-mlst-16.txt
- Also the proposal includes already widely accepted and used SIZE and MDTM commands
- The proposal is to adopt the IETF draft as part of GridFTP

Issue: packed transfers of large numbers of small files

- Often it is more efficient to send many files as single tar file
- More generally, it is desired to be able to plug in custom modules on server side
- This may have been provisioned by ESTO/ERET

Issue: Flexible striping

- There should be a way to dynamically and flexibly choose striping strategy

What are our next steps ?

- GGF8: Finalize the list
- Shortly after GGF8: submit the list as GGF document
- Between GGF8 and GGF9:
  - Propose, discuss, choose solutions
  - Prototype solutions
- GGF9: discuss draft of GridFTP v2.0 document
- By GGF10: have GridFTP v2.0 document ready,
  - Have working implementations